COG: COnsistent data auGmentation for object perception

Zewen He*[1,2][0000−0002−4782−3165], Rui Wu[3], and Dingqian Zhang[2][0000−0001−8378−4960]

[1] Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Computer and Control Engineering, University of Chinese Academy of Science, Beijing, China
[3] Horizon Robotics, Beijing, China
hezewen2014@ia.ac.cn

Abstract. Recently, data augmentation techniques for training convnets emerge one after another, especially focusing on image classification. They're always applied to object detection without further careful design. In this paper we propose COG, a general domain migration scheme for augmentation. Specifically, based on a particular augmentation, we first analyze its inherent inconsistency, and then adopt an adaptive strategy to rectify ground-truths of the augmented input images. Next, deep detection networks are trained on the rectified data to achieve better performance. Our extensive experiments show that our method COG's performance is superior to its competitor on detection and instance segmentation tasks. In addition, the results manifest the robustness of COG when faced with hyper-parameter variations, etc.

1 Introduction

Over the past two decades, the vision community has made considerable progress on object perception, including image classification[1], object detection[2] and instance segmentation[3]. This is mainly due to the emergence of deep convolutional neural networks(CNNs) and massive annotated data. Along with the increase of CNNs' capacity (including depth, width etc), accuracies of theses tasks continue to increase. However, this growth may bring catastrophic overfitting phenomenon. In order to improve CNN's generalization and robustness, data augmentation strategies are often used to generate data with more diverse input distribution.

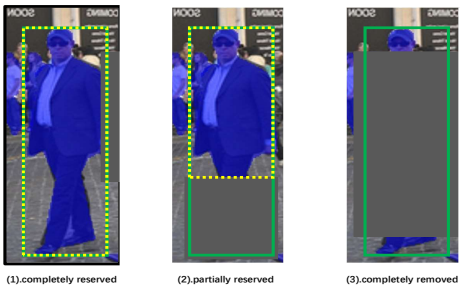
Existing augmentation techniques mainly stem from the image classification community. They usually contain three categories: 1). spatial transformation, such as random scale, flip, rotation; 2). color distortion, such as randomly changing pixel value in color space like RGB or HSV; 3). information dropping or stitching, such as randomly dropping regions; These methods try to change different properties of original images to obtain more training data, and then achieve considerable accuracy improvement.

Nevertheless, most augmentation methods are first proposed for the image classification task. They are always transferred to other tasks with little change. This direct migration is not quite reasonable. Because image classification just inputs one image and outputs a single label. The robustness requires that even if randomly changing input, the output label should be the same. But this requirement is not appropriate for other tasks, such as object detection and instance segmentation, which needs precise localization. For example in the detection task, if the augmentation randomly removes the half body of one annotated person from the image(please see Fig 2d), we need to figure out which bounding-box(bbox) matches the augmented input most consistently. The possible choices contain 1). original person box; 2). the person with half body; 3). the person disappears. The **first choice** will force the model trying to detect objects under severe occlusion, but this may be difficult if the whole person is occluded; The **second choice** wants to force the model obtaining accurate boundaries of objects, but this may fetch ambiguity of person category (the whole body and the half body both belong to the person category); The **third choice** just goes over to the opposite side of the first choice, namely deleting the ground-truth of objects (as ignored when training) which are under severe occlusion. In practice, the annotator also needs to make choices when meeting crowd scenes with occlusion. As far as we observe, most annotation in COCO[4] and CrowdHuman[5], characterize the whole outline of each object, namely the 1st choice. It's because the occlusion here is mainly from other foreground objects like horses or cars that can provide context information for predicting occluded targets. However, this is not the optimal choice for data augmentation. For example, GridMask[6] also makes the 1st choice as shown in fig 2e, but this will bring inconsistency if 90% of the object is occluded by the gray areas. The gray areas don't bring extra context information. Therefore, more intelligent choices are needed here.

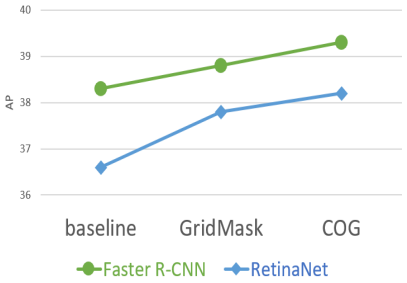
We believe that the key factor of making optimal choices is making the pair, namely (input image, output box, and label), consistent. If the occlusion is a little, the original gt-bbox can be reserved; If the occlusion is not severed but cannot be ignored, gt-bbox can be rectified; If the occlusion is very severe, the original gt-bbox can be removed. In this way, the best strategy is making the choice adaptively based on the degree of occlusion. Fig 1a shows different choices under different occlusion levels. To validate our hypothesis, we start from one augmentation method (GridMask) originated from the classification domain, and then point out the inconsistency when migrating to the detection domain directly. Next, we propose the COG method to rectify the inconsistency adaptively. The general difference are shown in Fig 2. Finally, COG can obtain higher performance in detection and instance segmentation tasks, which means that better choices are made in COG when facing region removal. The performance shown in fig 1b validates the superiority of COG.

In short, our main contribution can be summarized as follows:

- we analyze the inconsistency between input images and corresponding labels when encountering data augmentation in the detection task.
- we propose COG to rectify this inconsistency adaptively.



(a) **COG v.s. GridMask**: Here the blue region is the gt-mask of this person. And the gray region means that **GridMask** method occludes some part. The green and solid line is the gt-bbox of this person used in **GridMask**. The yellow and dash line is the gt-bbox of this person used in **COG**. According to the occlusion level, **COG** provides gt-bbox adaptively.



(b) **mAP on test-dev**: For both Faster R-CNN and RetinaNet, **COG** is superior to **GridMask**.

– we conduct extensive experiments and validate that our method benefits different detectors under various settings. Besides, comparable improvements have been achieved in other perception tasks.

The rest of this paper is organized as follows. Section 2 presents some related works about object detection and corresponding data augmentation. Section 3 analyzes the inconsistency and proposes COG. Experiment studies, including a comparison of the results and corresponding analysis, are presented in section 4. Finally, we conclude in section 5.

2 Related Work

We will introduce general data augmentation paradigms used in training CNN models. Next object detection and specially designed augmentation methods are also described.

2.1 Data Augmentation

Regularization is an effective technique to prevent CNN from over-fitting. Data augmentation is a special regularization which only operates on the data. It aims to increase the diversity of input distribution and is also easy to deploy. The basic augmentation policy consists of random flipping, random cropping, and random coloring, etc. Based on these policies, AutoAugment[7] tries to search the optimal combination of existing augmentations in virtue of reinforcement learning. [8,?] accelerates the searching process of AutoAugment. In addition, there are some methods that focus on deleting information in input

images through certain policies to strengthen the robustness. For example, random erasing[10] and cutout[11] randomly delete one continuous region in the image. Hide-and-Seek[12] divides the image into small patches and delete them randomly. GridMask[6] wants to drop and reserve information uniformly in images. Most methods previously mentioned are effective in training CNN models, but they are always experimented on the image classification task.

2.2 Object Detection

The object detection task attempts to locate and classify possible targets in the image at the same time. Benefit from the representation capacity of deep convfeature, CNN-based detectors [13,**?**,**?**,**?**,**?**] have become a dominant paradigm in the object detection community. The R-CNN series and its variants [2,**?**,**?**] gradually increase the upper bound of the performance on two-stage detectors. In particular, Faster R-CNN[2] is the principal architecture in these methods. It adopts a shared backbone network to extract features for subsequent proposal generation and RoI classification, resulting in real-time detection and rising accuracy. Besides, one-stage methods like RetinaNet[18] also work well.

Together with these efficient detectors, particular augmentation strategies are also designed for detection applications. Mosaic[19] tries to mix 4 training images with different contexts. This strategy significantly reduces the need for a larger mini-batch size. GridMask[6] can be extended to detection without special modification. Deep reinforcement learning is also used in [20] to find a set of best strategies for object detection automatically. InstaBoost[21] boosts the performance on instance segmentation by probability map guided copy-pasting techniques.

3 Method

We will formally introduce our COG, namely COnsistent auGmentation, in this section. To facilitate understanding, 3.1 analyzes the inconsistency in original GridMask. Then, 3.2 details the COG paradigm and its implementation.

3.1 Inconsistency in GridMask

Original GridMask Original GridMask[6] is a simple, general, and efficient strategy for data augmentation. Given an input image, GridMask randomly removes some regions which are distributed across the image uniformly. In other words, the removed regions are neither a continuous region[11] nor random pixels in dropout. They are disconnected pixel sets which are aligned to grids, as illustrated in Fig 2d. These gray regions with grid shape guarantee that both information deletion and reserve co-exist in augmentation.

In detail, the operation of GridMask can be summarized by eq 1.

$$\hat{\mathbf{x}} = \mathbf{x} \times \mathbf{M} \tag{1}$$

Concretely, $\mathbf{x} \in R^{H \times W \times C}$ denotes the original input image, $\mathbf{M} \in \{0,1\}^{H \times W}$ denotes the corresponding binary mask matrix, and $\hat{\mathbf{x}} \in R^{H \times W \times C}$ is the augmentation result generated by GridMask. For the binary mask $\mathbf{M}$, if $\mathbf{M}_{i,j} = 1$, we keep the original pixel (i, j) in $\mathbf{x}$; otherwise the pixel value at position (i, j) will be set to the mean RGB value, namely 0 value after image normalization.

Fig 2c displays the $\mathbf{M}$ with grid-shape and corresponding control parameters. It should be noted that the dark gray areas denote the regions where $\mathbf{M}_{i,j} = 0$; while the light white areas denote the regions where $\mathbf{M}_{i,j} = 1$. The parameters here contains $(r, d, \delta_x, \delta_y)$, which control the exact appearance of $\mathbf{M}$. Among them, d means the length of one grid unit; $1 - r$ means the proportion of the removed gray square in one grid unit; δ_x, δ_y means the start pixel of the first grid unit in an image.

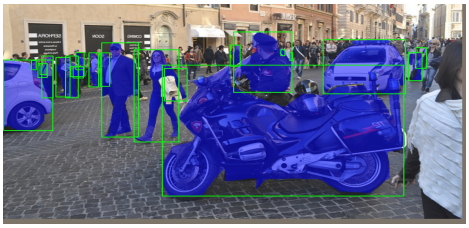
In order to increase the diversity of $\mathbf{M}$, these hyper-parameters will be generated randomly for each image. Unless otherwise specified, the d is sampled from $[d_l, d_h] = [32, 512]$, δ_x, δ_y is sampled from $[0, d - 1]$, and the r is set to 0.5 like original GridMask[6].

Inconsistency Details Although the original GridMask can increase the robustness of the detection model and improve the performance by nearly 1.0 points on COCO[4], there also exists inconsistency. The main inconsistency originates from mismatching between $\hat{\mathbf{x}}$, namely input image after GridMask, and the adopted ground-truth $gt(\mathbf{x})$ (including category labels and geometry bboxes in this image). If there is no augmentation, the input image $\mathbf{x}$ and corresponding ground-truth $gt(\mathbf{x})$ match perfectly. Because the $gt(\mathbf{x})$ is from the official annotation. But when GridMask is employed on some $\mathbf{x}$ to get $\hat{\mathbf{x}}$, some region in $\mathbf{x}$ is removed. At this moment, parts of some objects may be covered by the gray regions. Then, the original annotations of these objects don't match to $\hat{\mathbf{x}}$ well, as shown in Fig 2e. In other words, the original GridMask uses the changed image $\hat{\mathbf{x}}$ and the original but fixed annotation $gt(\mathbf{x})$ to train the CNN model. This strategy can be explained as increasing the robustness for occlusion, as the **first choice** mentioned in section 1. The CNN model will be trained to infer the whole object's category and geometry bbox when given a part of this object.

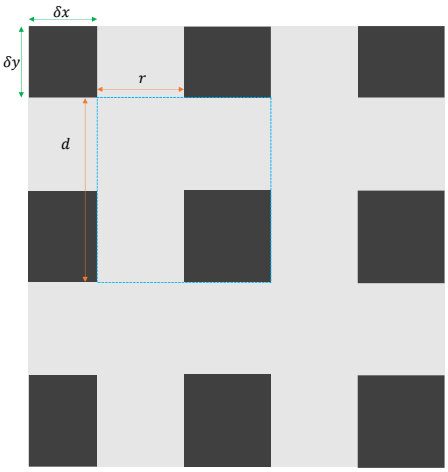
We argue that this approach may be challenged when the occlusion is severe in GridMask in both qualitative and quantitive perspectives. Qualitatively, the occlusion is inevitable as shown in Fig 2e. Even as an annotation worker, such as MTurk, he/she may feel difficult to give precise bounding-box. Quantitively, our statistical result shows that about 25.1% area of the foreground gt-masks are occluded when using default GridMask settings. This occlusion level cannot be ignored. Because we think that both the object itself and context features are important for object localization and recognition. If the occlusion is severe, the available information only stems from context. This may lead to ambiguity because the same context may contain different objects.

Intuitively, fig 1a illustrates the same gt-bboxes used in GridMask under different occlusion levels. Even if the whole person is almost completely occluded,

(a) Original image

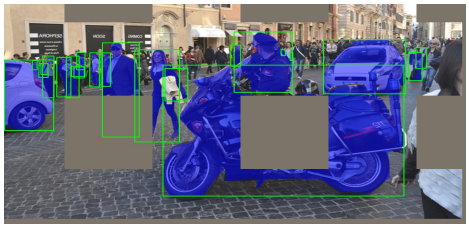


(b) Original image with ground-truth

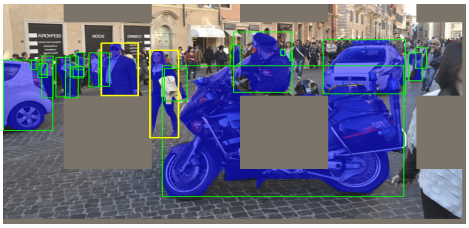


(c) Mask used in GridMask/COG

(d) Image after GridMask/COG



(e) GridMask image with ground-truth: all bboxes are marked by green rectangles.



(f) COG image with ground-truth: changed bboxes are marked by yellow rectangles.

Fig. 2: **Original v.s. GridMask v.s. COG** Blue regions represent the gt-mask. Green rectangles represent the gt-bbox.

GridMask still gives it a bounding-box annotation for training. This is unreasonable and inconsistent.

3.2 Rectifying Ground-Truth

As previously mentioned, there exists inconsistency between changed input image $\hat{\mathbf{x}}$ and the unchanged ground-truth $gt(\mathbf{x})$. We should rectify original ground-

truth $gt(\mathbf{x})$ to $gt(\hat{\mathbf{x}})$ and try to make the matching degree between $\hat{\mathbf{x}}$ and $gt(\hat{\mathbf{x}})$ as high as possible. In other words, we should find better choice from the three candidates adaptively, as section 1 mentioned.

The rectifying procedure in our method is displayed in **Algorithm** 1 In detail, for original input image ($\mathbf{x}$, whose corresponding ground-truth is $gt(\mathbf{x})$. There're N annotated objects in $gt(\mathbf{x})$. For each annotated object obj_k, there exists one category label c_k, one bounding box b_k, and one mask annotation $\mathbf{GM}_k$. Here, $\mathbf{GM}$ means the ground-truth mask for the object. As shown in Fig 2b, the b_k and $\mathbf{GM}_k$ are the green bbox and blue region respectively. Specificly in COCO dataset, the c_k is $0-1$ vector with 80 dimension; the b_k is (x_k, y_k, w_k, h_k), which represents the (x, y) coordinate of the top-left corner, width and height of b_k; the $\mathbf{GM}_k$ is also a binary mask matrix which shape is $H \times W$. If the pixel at (i, j) in x is in fore-ground region of b_k, then $\mathbf{GM}_k(i, j)$ is 1, otherwise 0. When GridMask operation is adopted, the binary mask $\mathbf{M}$ is shown in Fig 2c.

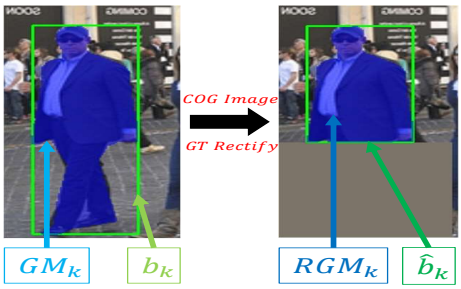


Fig. 3: **Rectifying Procedure**: Left fig represents original image and corresponding ground-truth: gtbbox b_k and gtmask $\mathbf{GM}_k$; Right fig represents image after GridMask operation and corresponding ground-truth after rectifying(when $thres_l < saveRatio_k < thres_h$): new gt-bbox $\hat{b}_k$ and gt-mask $\mathbf{RGM}_k$.

The core idea of **Algorithm** 1 is making a perfect tradeoff between object and context when rectifying. Concretely, COG calculates the reserved region $\mathbf{RGM}$ of each b_k under $\mathbf{M}$, and then provide a revised $\hat{b}_k$. As shown in Fig 3, the blue region at left is original $\mathbf{GM}$, and the blue region at right is $\mathbf{RGM}$. All $\hat{b}_k$ constitute the new ground-truth for $\hat{\mathbf{x}}$. In detail, according to the $saveRatio$, namely the ratio between $area(\mathbf{RGM})$ and $area(\mathbf{GM})$, different strategies will be used.

First, if $saveRatio_k$ is higher than $thres_h$, the $\hat{b}_k$ is the same as b_k. We consider this object as unbroken but just occluded, the object information doesn't disappear. Second, if $saveRatio_k$ is lower than $thres_h$ but higher than $thres_l$, the $\hat{b}_k$ is the minimum enclosing rectangle of $\mathbf{RGM}_k$. We consider this object as broken but still existent, the image has dropped part of the object information. The remanent object information and context information can be used to predict the object category but no precise localization for the original object. So

Algorithm 1 Rectifying Ground-Truth

Input: $\mathbf{M}, GT = \{(b_k, \mathbf{GM}_k), where \quad k \in [1, ..., N]\}$
Output: $\hat{GT}_{bbox}, \hat{GT}_{mask}$
0: $\hat{GT}_{bbox} \leftarrow \varnothing, \hat{GT}_{mask} \leftarrow \varnothing,$
1: **For** $k \in [1, ..., N]$ **do**
2: $\quad \mathbf{RGM}_k = \mathbf{M} \times \mathbf{GM}_k$
3: $\quad saveRatio_k = area(\mathbf{RGM}_k)/area(\mathbf{GM}_k)$
4: $\quad$ **if** $saveRatio_k > thres_h$
5: $\quad\quad \hat{b}_k = b_k$
6: $\quad\quad \hat{\mathbf{GM}}_k = \mathbf{GM}_k$
7: $\quad$ **elseif** $saveRatio_k > thres_l$
8: $\quad\quad \hat{b}_k = minEncloseRectangle(\mathbf{RGM}_k)$
9: $\quad\quad \hat{\mathbf{GM}}_k = \mathbf{RGM}_k$
10: $\quad$ **else**
11: $\quad\quad$ continue
12: $\quad \hat{GT}_{bbox} \leftarrow \hat{GT}_{bbox} \cup \{\hat{b_k}\}$
13: $\quad \hat{GT}_{mask} \leftarrow \hat{GT}_{mask} \cup \{\hat{\mathbf{GM}}_k\}$
14:Output $\{\hat{GT}_{bbox}, \hat{GT}_{mask}\}$ as new ground-truth for augmented image

we acknowledge that the object still exists but the corresponding gt-bbox should be changed to the truncated version. Third, the $\hat{b}_k$ will disappear. We consider that this object is occluded by $\mathbf{M}$ severely. The remanent object information is too little to predict the object category. Fig 1a illustrates corresponding results under different occlusion levels, namely $saveRatio_k$. It also explains COG's superiority intuitively.

It should be pointed out that the rectifying operation only changes ground-truth in accordance with the occlusion situation in the current image. So if $thres_h$ and $thres_l$ are set to 0.0, then the generated $\hat{GT}$ is the same as GT, namely the original GridMask. Besides, although it now uses $\mathbf{GM}_k$ to calculate $saveRatio_k$, it can also only use b_k to get a course rectification. More details about this will be described in 4.3.

4 Experiments

Extensive experiments are conducted on object detection and instance segmentation to verify the effectiveness of COG. As described below, we firstly depict common settings in detail for a fair comparison. Then, the main results and ablation study are provided to validate COG's advantage over other competitors and robustness to hyper-parameter variation.

4.1 Common Settings

Dataset Description Experiments are performed on COCO[4] following the official dataset split. In other words, all models are trained on train2017(118k

images) and evaluated on val2017(5k images). The results on test-dev(20k+ images) are submitted to the official server for evaluation.

Common Implementation and Hyper-parameters Two-stage Faster R-CNN[2] is employed for detection, and Mask R-CNN[3] is employed for instance segmentation. We implements our method and competitors based on MMDetection[22] framework. The details will be described from input images to outputs. In our experiments, first, input images are always resized to a single scale $(800, 1333)$ in both training and testing phases. Second, backbones in detection models generally include vanilla ResNet[1](abbr. R50 for ResNet-50). After that, FPN is adopted to extract features of multiple resolutions. For post-processing, Non-Maximum Suppression (NMS) is substitute by Soft-NMS[23] to remove possible duplicate bboxes.

Besides, all models are trained on 8 GPUs for 24 epochs. The learning rate is initialized with$0.00125*$batch-size with a gradual warmup strategy, then divided by 10 at 16-th and 22-th epoch successively.

Hyper-parameters in COG Without specification, the upper bound and lower bound are set to $[0.25, 0.9]$ in COG. The probability of augmentation operation is set to a constant value, namely 0.7, for both GridMask and COG. We also set $drange = [32, 512]$ and $r = 0.5$ for both COG and GridMask, as 3.1 mentioned.

Competitor methods Except for self-comparison, our proposed COG is compared with GridMask[6] which belongs to current SOTA methods.

Evaluation metrics Standard COCO metrics[4], including AP (mean AP over multiple IoU thresholds) for object detection, AP^{mask} for instance segmentation are reported. Note that the best results in each table are in **boldface**.

4.2 Main Results

To verify the effectiveness of COG, we compare COG with GridMask in different detectors (such as Faster RCNN and RetinaNet), when integrated with FPN. Table 1 shows the results on COCO val dataset. For baseline Faster RCNN with R50 backbone, our re-implemented baseline AP is 38.0, which surpasses that in [22] because all models are equipped with Soft-NMS. Compared with the baseline , while GridMask improves the AP by 0.6 points, COG boosts additional 0.5 points further. Also for RetinaNet, while GridMask improves the AP by 0.9 points, COG boosts additional 0.4 points further. The same performance improvement can be observed in the COCO test-dev dataset, as shown in table 2. We can conclude that COG is superior to GridMask in the detection task.

Among them, the accuracy improvement is higher in AP_M and AP_L, compared with AP_S. First, we argue that because the object with a smaller scale will be occluded by the gray region with higher probability. So more small objects are removed in the training phase and relatively more medium or large objects participate in the training process. Second, when more medium or large objects are reserved, the rectified gt-bbox also diminishes the inconsistency in the original GridMask.

Table 1: Detection Results on COCO val

Method	Backbone	AP	AP_S	AP_M	AP_L
FPN**(baseline)**	R50	38.0	22.0	41.7	49.3
GridMask+FPN	R50	38.6(+0.6)	22.7	42.4	49.2
COG+FPN	R50	**39.1(+1.1)**	**23.0**	**43.1**	**49.7**
RetinaNet**(baseline)**	R50	36.3	19.5	39.7	47.6
GridMask+RetinaNet	R50	37.2(+0.9)	21.3	41.2	48.5
COG+RetinaNet	R50	**37.6(+1.3)**	**21.5**	**41.7**	**48.9**

Table 2: Detection Results on COCO test-dev

AP	AP_S	AP_M	AP_L
38.3	21.9	41.3	47.6
38.8(+0.5)	22.5	42.0	47.6
39.3(+1.0)	**22.9**	**42.6**	**48.1**
36.6	19.6	39.4	46.1
37.8(+1.2)	21.3	41.1	47.2
38.2(+1.6)	**21.5**	**41.7**	**47.9**

4.3 Ablation Study

Hyperparameter Settings In this section, results of Faster R-CNN with COG under different hyper-parameters are displayed in table 3.

Firstly, we experiment with the upper and lower bound of *saveRatio*, namely $thres_h$ and $thres_l$, on COG+FPN. We need to reserve the original annotation when occlusion is not severe to get robustness, We also need to reserve the visible section or delete it when occlusion is severe. So the upper bound cannot be very high or very low. Similarly, the lower bound cannot be very low. According to the result, the upper bound $thres_h = 0.9$ and lower bound $thres_l = 0.25$ performs best. Secondly, we experiment with different *drange*, namely $[d_l, d_h]$ mentioned in subsection 3.1, on COG and original GridMask. It seems that the expanded range for d is not good for GridMask but okay for COG.

These variations show the robustness of the COG paradigm.

Method	drange	maskratio	AP
COG+FPN	[32, 512]	[0.25, 0.5]	38.7
COG+FPN	[32, 512]	[0.25, 0.75]	38.8
COG+FPN	[32, 512]	[0.25, 0.9]	**39.1**
COG+FPN	[32, 512]	[0.25, 1.0]	38.8
COG+FPN	[32, 512]	[0.3, 0.9]	39.0
COG+FPN	[32, 512]	[0.4, 0.9]	38.7
COG+FPN	[2, 800]	[0.25, 0.9]	38.9
GridMask+FPN	[2, 800]	[0.25, 0.9]	38.5

Table 3: Detection Results of COG under different hyper-parameters. This demonstrates that COG is robust to different settings. Details are explained in subsection 4.3.

Method	step_list	prob_list	AP
COG+FPN	[,24]	[,0.7]	**39.1**
COG+FPN	[2,24]	[0.0,0.7]	39.0
COG+FPN	[1,2,4]	[0.0,0.7,0.9]	39.0
COG+FPN	[23,24]	[0.7,0.3]	38.9
Changing loss weight	RPN	RCNN	
COG+FPN	True	True	38.8
GridMask+FPN	True	True	38.5
Using bbox ratio	drange	bboxratio	
COG+FPN	[32, 512]	[0.3, 0.8]	39.0
COG+FPN	[32, 512]	[0.3, 0.9]	38.8

Table 4: Detection Results of COG with other variations. This demonstrates that COG is robust to different training settings. Details are explained in subsection 4.3.

Variations of COG We also experiment with more variations on COG to validate its robustness.

The first variation is changing the probability of augmentation. For example in table 4, step_list=$[2, 24]$ and prob_list=$[0.0, 0.7]$ means setting COG's proba-

bility to 0.0 before the 2-th epoch and setting this probability to 0.7 between 2-th and 24-th epoch. We can see that the performance of COG is consistent regardless of the variation of COG's probability. The second variation is changing the loss weight of k_{th} gt-bbox by the $saveRatio_k$ adaptively but adopting the original ground-truth in training. If the $saveRatio_k$ is low, then the corresponding loss weight also decreases. It can be regarded as a soft version of ground-truth rectification. From the results in table 4, we can see that adaptively changing loss weight doesn't make a difference for COG and GridMask. The third variation is using bbox save ratio but not mask save ratio. In detail, the save ratio is $saveRatio_k = \text{Area}(bb_k^{res})/\text{Area}(bb_k)$. Here, bb_k means the k_{th} gt-bbox, bb_k^{res} means the maximum rectangle that hasn't been occluded by GridMask in bb_k. From the results in table 4, we can see that only using bbox annotation for calculating save ratio in COG can also achieve similar performance.

COG on Instance Segmentation We also experiment with COG on the instance segmentation task to validate its effectiveness on rectifying gt-mask annotations. As shown in table 5, COG surpasses GridMask 0.4 points on AP and 0.2 points on Mask AP.

Table 5: Mask RCNN's results on COCO val

Method	Backbone	AP	AP_S	AP_M	AP_L	AP^{mask}	AP_S^{mask}	AP_M^{mask}	AP_L^{mask}
FPN**(baseline)**	R50	39.1	22.3	42.5	51.0	34.8	18.5	37.9	47.8
GridMask+FPN	R50	39.2	23.3	43.1	50.2	35.2	18.8	38.7	47.4
COG+FPN	R50	**39.6**	23.3	**43.6**	**50.6**	**35.4**	**18.9**	**38.9**	**48.0**

5 Conclusion

In this paper, we propose COG, an adaptive rectification strategy for data augmentation, which eliminates the inherent inconsistency. The experimental studies validate that COG can improve data augmentation's performance on different perception tasks. It's also robust to various settings for hyper-parameters and training configurations. COG provides a new perspective to migrate data augmentation from label-based domain(classification) to location-based domain (detection). Further, we can extend COG to considering both image and current network's state simultaneously.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of CVPR. (2016) 770–778
2. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of NIPS. (2015) 91–99

3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of IEEE ICCV. (2017) 2980–2988
4. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: Proceedings of ECCV. (2014) 740–755
5. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123 (2018)
6. Chen, P., Liu, s., Hengshuang, Z., Jiaya, J.: Gridmask data augmentation. arXiv preprint arXiv:2001.04086 (2020)
7. Cubuk, E.D., Zoph, B., Mané, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. Proceedings of IEEE CVPR (2019) 113–123
8. Lim, S., Kim, I., Kim, T., Kim, C., Kim, S.: Fast autoaugment. arXiv preprint arXiv:1905.00397 (2019)
9. Hataya, R., Zdenek, J., Yoshizoe, K., Nakayama, H.: Faster autoaugment: Learning augmentation strategies using backpropagation. arXiv preprint arXiv:1911.06987 (2019)
10. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: AAAI. (2020)
11. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
12. Singh, K.K., Hao, Y., Sarmasi, A., Pradeep, G., Yongjae, L.: Hide-and-seek: A data augmentation technique for weakly-supervised localization and beyond. (2018)
13. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of IEEE CVPR. (2014) 580–587
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: single shot multibox detector. In: Proceedings of ECCV. (2016) 21–37
15. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
16. Li, Y., Chen, Y., Wang, N., Zhang, Z.: Scale-aware trident networks for object detection. arXiv preprint arXiv:1901.01892 (2019)
17. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: Proceedings of CVPR. (2018) 6154–6162
18. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of IEEE ICCV. (2017) 2999–3007
19. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
20. Zoph, B., Cubuk, E.D., Ghiasi, G., Lin, T.Y., Shlens, J., Le, Q.V.: Learning data augmentation strategies for object detection. arXiv preprint arXiv:1906.11172 (2019)
21. Fang, H.s., Sun, J., Wang, R., Gou, M., Li, Y., Lu, C.: Instaboost: Boosting instance segmentation via probability map guided copy-pasting. Proceedings of IEEE ICCV (2019) 682–691
22. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
23. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms improving object detection with one line of code. In: Proceedings of IEEE ICCV. (2017) 5562–5570